

I. CLAIM AMENDMENTS

1. (currently amended) A computer implemented method for an analysis of a plurality of software components of a plurality of applications prior to deployment of the plurality of applications, comprising:

using a data structure in a storage that provides, for each of the plurality of software components from the plurality of applications, a component deployment dependency data, an indication of necessary components for an operation of each of the plurality of software components being installed, and an indication of incompatibility with a previously installed component; and

using a computer connected to the storage and a program installed in a memory of the computer, performing the steps of:

determining a first plurality of software components previously installed on a system;

determining a second plurality of software components to be installed on the system;

accessing a third plurality of component deployment dependency data;

determining a fourth plurality of software components suitable for parallel installation;

determining an order in which the fourth plurality of software components can be grouped for a fifth plurality of parallel installations;

accessing a sixth plurality of metadata from the data structure regarding the second plurality of software components to be installed and accessing a seventh plurality of metadata regarding the first plurality of software components previously installed; and

analyzing the sixth plurality of metadata to determine an eighth plurality of potential conflicts between the second plurality of software components to be installed and the first plurality of software components previously installed on the system;

wherein the pre-deployment analysis allows the second plurality of software components to be installed in parallel and in a sequence of groups;

wherein an installation time for the plurality of applications is reduced; and

wherein the plurality of ~~software components comprises an IBM® Websphere Application Server (WAS)~~ applications include an application server.

2. Cancelled.

3. (currently amended) The computer implemented method of claim 1, further comprising updating the data structure with an identity of a ninth plurality of recently installed software components.

4. (currently amended) The computer implemented method of claim 1, further comprising providing a user with a plurality of options for the eighth plurality of potential conflicts.

5. (currently amended) The computer implemented method of claim 4, wherein a first option includes aborting an installation.

6. (currently amended) The computer implemented method of claim 4, wherein a second option includes continuing an installation.

7. (currently amended) The computer implemented method of claim 6, further including, upon the exercise of the second option, recording an entry in a log indicative of a conflict and of a continuation of installation.

8. (currently amended) The computer implemented method of claim 1, further comprising:
initiating a removal of a software component from the system; and
identifying a tenth plurality of remaining software components which depend on the software component to be removed.

9. (currently amended) The computer implemented method of claim 8, further comprising providing a user with a plurality of options if the tenth plurality of dependent remaining software components is identified.

10. (currently amended) The computer implemented method of claim 9, wherein a first option includes aborting a removal.

11. (currently amended) The computer implemented method of claim 9, wherein a second option includes continuing a removal.

12. (currently amended) The method of claim 8, further comprising:

identifying a first software component previously installed on the system which is dependent upon a removed software component; and

determining an identity of a second software component upon which the first software component depends.

13. (currently amended) The method of claim 12, further comprising:

installing the second software component upon which the first software component depends; and

creating a dependency link between the first software component and the second software component.

14. (currently amended) A system for an analysis of a plurality of software components of a plurality of applications to be conducted before installing the plurality of ~~software~~ components~~applications~~, comprising:

using a data structure in a storage that provides, for each of the plurality of software components, a component deployment dependency data, an identification of necessary components for an operation of each of the plurality of software components to be installed, and an identification of incompatibility with a previously installed software component;

using a computer connected to the storage and to a system; and

using a program installed in a memory of the computer, the program comprising:

means for determining a first plurality of software components previously installed on the system;

means for determining a second plurality of software components to be installed on the system;

means for accessing a third plurality of software component deployment dependency data;

means for determining a fourth plurality of software components suitable for parallel installation;

means for determining an order in which the fourth plurality of software components can be grouped for a fifth plurality of parallel installations;

means for accessing a sixth plurality of metadata from the data structure regarding the second plurality of software components to be installed and accessing a seventh plurality of metadata regarding the first plurality of software components previously installed; and

means for analyzing the sixth plurality of metadata to determine an eighth plurality of potential conflicts between the second plurality of software components to be installed and the first plurality of software components previously installed on the system;

wherein a pre-deployment analysis allows the second plurality of software components to be installed in parallel and in a sequence of groups;

wherein an installation time for the plurality of applications is reduced; and

wherein the plurality of ~~software components comprises~~ applications includes an IBM[®] Websphere Application Server (WAS) application server.

15. Cancelled.

16. (currently amended) The system of claim 14, further comprising a means for loading an installation package including the data structure.

17. (currently amended) The system of claim 14, further comprising a ninth plurality of references among the plurality of software components to be installed and located in the data structure.

18. (previously presented) The system of claim 17, further comprising a means for accessing the data structure.

19. (currently amended) The system of claim 14, further comprising a means for installing the second plurality of software components across a plurality of enterprise resources.

20. (currently amended) A data structure encoded in a computer-readable ~~media~~ medium and associated with a software component installation package adapted for execution on a computer, the data structure adapted for identifying a third plurality of potential conflicts between a second plurality of software components to be installed on a system and a first plurality of software components previously installed on the system, comprising:

for each of the second plurality of software components, a component deployment dependency data, an indication of necessary components for an operation of each of the second plurality of software components, and an indication of incompatibility with one or more software components of the first plurality of software components;

wherein an alert is automatically generated if an attempt is made to install a software component having an indication of incompatibility; and

wherein the software component installation package is adapted for installation of an ~~IBM® Websphere Application Server (WAS)~~ application server.

21. (currently amended) A computer program product of a computer-readable medium usable with a programmable computer, the computer program product having computer-readable code embodied therein for pre-deployment analysis of a plurality of software components of a plurality of applications, the computer-readable code comprising instructions for:

determining a first plurality of software components previously installed on a system;

determining a second plurality of software components to be installed on the system;

accessing a third plurality of software component deployment dependency data; determining a fourth plurality of software components suitable for parallel installation;

determining an order in which the fourth plurality of software components can be grouped for a fifth plurality of parallel installations;

accessing a sixth plurality of metadata from a data structure regarding the second plurality of software components to be installed and accessing a seventh plurality of metadata regarding the first plurality of software components previously installed; and

analyzing the sixth plurality of metadata to determine an eighth plurality of potential conflicts between the second plurality of software components to be installed and the first plurality of software components previously installed on the system;

wherein the pre-deployment analysis allows the second plurality of software components to be installed in parallel and in a sequence of groups;

wherein an installation time for the plurality of applications is reduced; and

wherein the ~~second plurality of components~~ applications includes an ~~comprises an IBM®~~
~~Websphere Application Server (WAS)~~ application server.

22. Cancelled.

23. (currently amended) The computer program product of claim 21, further comprising instructions for updating the data structure with an identity of a ninth plurality of recently installed software components.

24. (original) The computer program product of claim 21, further comprising instructions for providing a user with a plurality of options if a conflict is identified.

25. (previously presented) The computer program product of claim 24, wherein a first option includes aborting an installation.

26. (previously presented) The computer program product of claim 24, wherein a second option includes continuing an installation.

27. (previously presented) The computer program product of claim 26, further including instructions for, upon the exercise of the second option, recording an entry in a log indicative of the conflict and of a continuation of the installation.

28. (currently amended) The computer program product of claim 21, further comprising instructions for:

initiating a removal of a software component from the system; and
identifying a plurality of remaining software components which depend on the software component to be removed.

29. (currently amended) The computer program product of claim 28, further comprising instructions for providing a user with a plurality of options if a dependent remaining software component is identified.

30. (original) The computer program product of claim 29, wherein a first option includes aborting the removal.

31. (original) The computer program product of claim 29, wherein a second option includes continuing the removal.

32. (currently amended) The computer program product of claim 28, further comprising instructions for:

identifying a first software component previously installed on the system which is dependent upon a removed software component; and

indicating the identity of a second software component upon which the first software component depends.

33. (currently amended) The computer program product of claim 32, further comprising instructions for:

installing the second software component upon which the first software component depends; and

creating a dependency link between the first software component and the second software components component.

34. (currently amended) A method for installing a plurality of software components from a plurality of applications based upon a pre-installation analysis, comprising:

loading an installation package, the installation package including a data structure;

searching a target to which the plurality of software components are to be installed to identify a plurality of previously installed software components;

for a first software component, accessing, in the data structure, a component deployment dependency data, an indication of necessary software components for an operation of the first software component, and an indication of incompatibility with a previously installed software component;

analyzing a plurality of data from the data structure to determine a plurality of conflicts between the first software component to be installed and the plurality of software components previously installed on the system; and

wherein the plurality of ~~software components~~ applications ~~comprises~~ includes an IBM® ~~Websphere Application Server (WAS)~~ application server.

35. (original) The method of claim 34, further comprising notifying a user of the conflict.

36. (original) The method of claim 34, further comprising aborting the installation if a conflict is detected.

37. (original) The method of claim 34, further comprising ignoring a detected conflict and continuing the installation.

38. (original) The method of claim 37, further comprising entering a note in a log of the conflict.

39. (currently amended) The method of claim 34, further comprising:

initiating the removal of an installed software component;

accessing the data structure; and

identifying a conflict if the installed software component is removed.

40. (currently amended) The method of claim 34, further comprising:

initiating an installation of a second software component;
searching a target to which the second software component is to be installed to
identify installed software components;
accessing the data structure; and
determining if all of the other software components required by the second
software component are installed.